Asymptotic hitting time for a simple evolutionary model of protein folding

Véronique Ladret

Laboratoire LaPCS, U.F.R. de Mathématiques,
Université Claude Bernard - Lyon 1, 50, avenue Tony Garnier,
Bâtiment RECHERCHE [B], Domaine de Gerland
69367 Lyon Cedex 07, France
email : veronique.ladret@univ-lyon1.fr

February 1, 2008

Abstract

We consider two versions of a simple evolutionary algorithm model for protein folding at temperature zero: the $(1+1)$-EA on the LeadingOnes problem. In this schematic model, the structure of the protein, which is encoded as a bit-string of length n, is evolved to its native conformation through a stochastic pathway of sequential contact bindings.

We study the asymptotic behavior of the hitting time, in the mean case scenario, under two different mutations: the one flip which flips a unique bit chosen uniformly at random in the bit-string, and the Bernoulli flip which flips each bit in the bit-string independently with probability c/n. For each algorithm we prove a law of large numbers, a central limit theorem and compare the performance of the two models.

KEY WORDS: evolutionary algorithm, markov chain, protein folding.
A.M.S. SUBJECT CLASSIFICATION : 60J10, 60F05, 92D20, 92C05.

1 Introduction

Evolutionary algorithms (EAs) are adaptive heuristic search algorithms. They are based on the mechanisms of natural selection and are widely used in a great variety of problems, for instance population genetics, machine learning and optimization. The task of the EA is to search a fitness landscape for maximal values. A population of individuals, considered as candidate solutions to the given problem, is evolved under steps of mutation and steps of selection. Each individual receives a numerical evaluation, called its fitness score. The dynamics of the EA simulates, supposedly like in natural systems, the survival of the fittest among the individuals. Thus, individuals of maximum fitness are sought.

Despite their numerous heuristic successes, mathematical results describing the behaviour of EAs are rather sparse. Among the exceptions are R. Cerf [9], [8], Y. Rabinovich and A. Wigderson [19], G. Rudolph [20] , C. Mazza and D. Piau [16], P. Del Moral and A. Guionnet [17], J. Bérard [4] and J. Bérard and A. Bienvenüe [5], [6].

Since EAs usually exhibit complicated dynamics, complexity results are difficult to reach and it is a common approach to consider simplified cases. Among the simplified EAs are the so called $(1+1)$-EAs. These are studied by H. Muhlenbein [18], T. Bäck [2], G. Rudolph [20], J. Garnier et al. [14], and S. Droste et al. [13], [12]. In this paper, we study the time of convergence of two versions of a specific $(1+1)$-EA, namely the $(1+1)$-EA on the LeadingOnes problem. One of the main motivation for studying these algorithms is that they can be used as a simple models for the protein-folding problem. Indeed, the $(1+1)$-EAs we focus on, directly fit to the model of protein-structure prediction at temperature zero, proposed by biophysicists A. Bakk et al. [3].

1.1 The physical model

Proteins typically fold to a unique native or biologically active conformation on time scales from 10^{-3}s. to 1s. However, if the dynamics of the folding process would follow a random search in the conformation space it would result in astronomical time scales. This paradox is known under the name of Levinthal's paradox [15]. How do proteins fold to their native state? This is one of the intriguing problems of biophysics. Anfinsen [1] showed that the native state is genetically as well as thermodynamically determined, i.e. it corresponds to the conformation in which Gibbs free energy of the whole system is lowest.

There are many hypotheses concerning the transition state (TS). One of the view is that the (TS)-dynamics consists in a pathway which carries the polypeptide (protein) to the native state through a guided descent along the Gibbs free energy landscape (J. A. Schellman [21],K. A. Dill et al. [10]).

The protein-like model proposed by A. Bakk et al. [3] can be described as follows. The polypeptide chain is equipped with n contact points $c_1, ..., c_n$ that we will also call nodes. For i from 1 to n, c_i is assigned a binary contact variable ϕ_i that indicates whether it is folded ($\phi_i = 1$) or unfolded ($\phi_i = 0$). In consequence the conformation of the protein is entirely determined by the bit-string of length n, $\phi = (\phi_1, \phi_2, ..., \phi_n)$ and the native state corresponds to the bit-string where $\phi_i = 1, \forall 1 \leq i \leq n$, that is $(1, ..., 1)$. There is a bijective mapping from the conformation space onto $\{0,1\}^n$.

Let i_0 denote the smalest $i \in \{1, ..., n\}$ for which c_i is unfolded, i.e. for which $\phi_i = 0$. We call the open part of the protein the set of contact points $\{c_{i_0}, c_{i_0+1}, ..., c_n\}$.

The assumption about the dynamics of the folding process is that each individual node is assigned an energy of $-\varepsilon_0$ if $i < i_0$, zero otherwise. It can be implemented trhough the

Hamiltonian

$$\mathcal{H} = -\varepsilon_0(\phi_1 + \phi_1\phi_2 + ... + \phi_1...\phi_n)$$

This Hamiltonian can be re-writen in terms of the LeadingOnes function L, defined on the space of conformations $\{0,1\}^n$, which counts the length of the longest prefix of ones in the bit-string:

$$L(x) = \max\{k \geq 1 : \forall 1 \leq i \leq k, x_i = 1\} \cup \{0\}.$$

Indeed,

$$\mathcal{H} = -\varepsilon_0 L(\phi) = -\varepsilon_0(i_0 - 1) \tag{1}$$

In this model, there is no energy associated to the open part of the protein. It is also known under the name of the "zipper-model". In fact, a descent along the energy landscape means both the folding of the leftmost uncorrectly folded substructure, i.e node c_{i_0}, and the "status-quo" for the correctly folded substructures preceding it (on the left), i.e. nodes c_i for $i < i_0$. It means that the folding events occur in a specific order: they behave like the individual locks in a zipper. In the bit-string framework, lowering the Gibbs free-energy is exactly increasing the size of the longest prefix of ones.

The algorithm proposed by A. Bakk et al. [3] to search the state space $\{0,1\}^n$ for conformations of lowest energy, i.e. the native state, is based on the Monte Carlo Metropolis (MCM) method (Binder [7]). Let T denote the temperature of the whole system, k the usual Boltzmann constant, and put $\beta = 1/kT$. The algorithm proceeds iteratively as follows. The individual (bit-string) at time k, X_k, undergoes a mutation to a new conformation X'_k, through a stochastic process that will be described later. Now, X'_k is selected to form the new individual at time $k+1$, X_{k+1}, with probability

$$\begin{aligned} P_{\text{accept}} &= \min\left(1, \exp(-\beta\Delta\mathcal{H})\right), \text{ with} \\ \Delta\mathcal{H} &= \mathcal{H}(X'_k) - \mathcal{H}(X_k). \end{aligned}$$

Otherwise X_{k+1} is a repeat of the old configuration X_k.

In this paper, we concentrate on the MCM model taken at temperature zero, that we denote by MCM$_0$, as well as on a very close version of this algorithm. These algorithms are directly connected to the $(1+1)$-EA on the LeadingOnes problem. We recall that the dynamics of $(1+1)$-EAs can be formalized through discrete Markov chains as follows:

1.2 The $(1+1)$-EA approach to native conformation prediction

In the protein model we wish to minimize the Hamiltonian $\mathcal{H}$, which is equivalent, according to (1), to maximize the LeadingOnes function. More generally, the goal of $(1+1)$-EAs is to optimise some fitness function $f : \{0,1\}^n \to \mathbb{R}$. The algorithm proceeds as follows: a unique individual, or bit-string, is evolved under the following two-steps iterative process:

1.**Mutation:**
As in the MCM method, at every evolutionary step, known as generation, the individual in the current population at time k, X_k, undergoes a random walk to a new individual X'_k.
2. Selection:
X_k and X'_k are evaluated in terms of their fitness value. Then, the one with the highest fitness score is selected to form the generation at time $k+1$, X_{k+1}:

$$\text{If } f(X'_k) > f(X_k), \text{ then } X_{k+1} = X'_k. \text{ Otherwise } X_{k+1} = X_k. \tag{2}$$

The notation (1+1)-EA accounts for the fact that we select among one parent and one child. Here, we focus on the mean case scenario, in which the first individual X_0 is chosen uniformly at random in $\{0,1\}^n$. The reason for this restriction is that the mean case is easier to manage, on a mathematical level, in the LeadingOnes framework. When the fitness function is precisely L, this algorithm will be denoted by $(1+1)_L$-EA. We notice that in the case of a fitness landscape with local maxima the $(1+1)$-EA method could end in a suboptimal search. But, as long as we consider the LeadingOnes problem whose fitness has no local maxima, we are not worried about that. In the litterature, there is no actual consensus, in the definition of $(1+1)$-EAs, on the selection rule. It is sometimes taken to be the following sligthly different one:

$$\text{If } f(X'_k) \geq f(X_k), \text{ then } X_{k+1} = X'_k. \text{ Otherwise } X_{k+1} = X_k. \tag{3}$$

For example, Garnier et al. [14] consider the first version of the selection rule (2), whereas Droste et al. [13], [12] focus the second version (3).

We notice that in the LeadingOnes framework, the variant of $(1+1)_L$-EA with mutation rule (3) is nothing else but MCM$_0$. In order to discriminate them, we shall keep the name MCM$_0$. The only difference between these two very close algorithms being that MCM$_0$ accepts candidates X'_k whose energy is the same as the one of X_k, whereas $(1+1)_L$-EA does not.

1.3 Statement of the results

T_n (respectively $\widehat{T_n}$) denotes the hitting time until some optimal (with regard to the fitness function) conformation or individual is sampled by the $(1+1)$-EA (by MCM$_0$ respectively). We focus on both MCM$_0$ and $(1+1)_L$-EA in the mean case scenario, under two different kinds of mutation: the one flip, which flips a unique bit chosen uniformly at random in the bit-string, and the Bernoulli flip, which flips each bit in the bit-string independently with probability c/n. As briefly recalled in the introduction, there has already been some work about the complexity of some $(1+1)$-EAs:
From Droste, Jansen and Wegener [13], $E(\widehat{T_n}) = \Theta(n \ln n)$ for the Bernoulli flip applied to a linear fitness function on $\{0,1\}^n$. From Droste et al. [12], the LeadingOnes function, is solvable in mean time $\Theta(n^2)$ in the Bernoulli flip scenario.

Garnier et al. [14] study the OneMax function $|\cdot|$, which counts the number of ones in the bit-string, in the one flip and the Bernoulli flip frameworks, that is

$$|x| = \sum_{i=1}^{n} x_i.$$

In the one flip case, $(T_n - n \ln n)/n$ converges in distribution to $-\ln 2 - \ln Z$. In the Bernoulli flip case, $(T_n - c^{-1}e^c n \ln n)/n$ converges in distribution to $-c^{-1}e^c \ln Z + C(c)$, where the law of Z is exponential of parameter 1 and $C(c)$ is some c-dependent constant.
We prove the analog for the LeadingOnes problem of the result of Garnier et al. [14]. This improves on the result of Droste et al. [12]. We prove a law of large numbers, a central limit theorem, and we compare the performance of the two models. Finally, we prove that the distribution of the hitting time of MCM_0, $\widehat{T}_n$, is the same as the one of T_n in both the one flip and the Bernoulli flip scenari.

Theorem 1.1 (one flip case) *(i) For $n \geq 1$, $E(T_n) = n^2/2$;*
(ii) As $n \to \infty$, $T_n/E(T_n)$ converges in probability to 1.
(iii) As $n \to \infty$, $(T_n - E(T_n))/n^{3/2}$ converges in distribution to a centered Gaussian random variable of variance 3/4.

Theorem 1.2 (Bernoulli flip case) *i) As $n \to \infty$, $E(T_n) \sim m(c)\, n^2$, with*

$$m(c) := (e^c - 1)/(2c^2).$$

ii) As $n \to \infty$, $T_n/E(T_n)$ converges in probability to 1.
iii) Furthermore, $(T_n - m(c)\, n^2)/n^{3/2}$ converges in distribution to a centered Gaussian random variable of variance $\sigma^2(c)$, with

$$\sigma^2(c) := 3(e^{2c} - 1)/(8c^3).$$

Note that $m(c) > 1/2$ for every $c > 0$.

Corollary 1.1 *As $n \to \infty$, $E(T_n)$ for the Bernoulli flip case is greater than $E(T_n)$ for the one flip case, for any value of c.*

Theorem 1.3 *In both the one flip and the bernoulli flip case, T_n and $\widehat{T}_n$ have the same distribution.*

2 Proof of Theorem 1.1

The law of T_n, conditioned by $|X_0|$, is the law of a sum of geometric random variables, see Lemma 2.1. This yields Part (i) of the theorem. Since the CLT implies the law of large numbers, we then prove the CLT of Part (iii).

The distribution of T_n stems from a simple observation. In the (1+1)-EA on the LeadingOnes problem, a mutation is accepted if and only if it adds 1 to the number of leading ones. As a consequence, in the one flip framework, the chain jumps when the leftmost zero is flipped. The other flips leave the chain unchanged. Thus, the zeroes in X_0 are successively flipped, from left to right, until one hits the optimal individal (conformation) $(1, 1, \ldots, 1)$.

Here and later in the paper, $\varepsilon = 1/n$ when we deal with algorithms on strings of length n, $|x|$ is the number of ones in x, the geometric law $\mathcal{G}(p)$ of parameter p is defined by

$$\mathcal{G}(p) = \sum_{n \geq 1} p\,(1-p)^{n-1} \delta_n,$$

and the negative binomial law of parameter (k_0, p), $\mathcal{NB}(k_0, p)$, puts the following mass on $k \geq k_0$:

$$\binom{k-1}{k-k_0} p^{k_0} (1-p)^{k-k_0}.$$

Lemma 2.1 *If $|X_0| = n - k_0$, T_n is the sum of k_0 i.i.d. $\mathcal{G}(\varepsilon)$ random variables. Thus, the law of T_n is negative binomial of parameter (k_0, ε).*

Proof of Lemma 2.1. Let $\tau_0 = 0$ and, for every $k \geq 0$,

$$\tau_{k+1} = \inf\{i \geq \tau_k\,; X_i \neq X_{\tau_k}\}, \quad \sigma_{k+1} = \tau_{k+1} - \tau_k, \quad \widetilde{X}_k = X_{\tau_k}. \tag{4}$$

In words, $\widetilde{X}_k$ denotes the position of the chain after its k-th jump, and $|\widetilde{X}_k| = n - k_0 + k$. The leftmost zero of $\widetilde{X}_k$ is flipped after a sojourn at $\widetilde{X}_k$ of length σ_{k+1}. Thus, $(\sigma_k)_k$ is i.i.d. of law $\mathcal{G}(\varepsilon)$. It remains to note that

$$T_n = \sigma_1 + \cdots + \sigma_{k_0}.$$

□

Proof of Part (iii). Let E_k denote the conditioning on $\{|X_0| = n - k\}$. Since X_0 is uniform, the law μ of $|X_0|$ is binomial $(n, 1/2)$. From Lemma 2.1, under P_k, T_n is the sum of k i.i.d. geometric random variables of parameter ε. Thus,

$$E_k(\mathrm{e}^{-\alpha T_n}) = \mathrm{e}^{-\alpha k}\, \varepsilon^k \left[1 - (1-\varepsilon)\,\mathrm{e}^{-\alpha}\right]^{-k}. \tag{5}$$

Set $\Theta_n = (T_n - n^2/2)/n^{3/2}$. The decomposition of the Laplace transform of Θ_n along the values of $|X_0|$, the explicit form of μ, and Equation (5) yield together that

$$E(\mathrm{e}^{-\alpha \Theta_n}) = \mathrm{e}^{\sqrt{n}\alpha/2}\, 2^{-n} \sum_{k=0}^{n} \binom{n}{k} \mathrm{e}^{-\alpha k/n^{3/2}}\, \varepsilon^k \left[1 - (1-\varepsilon)\,\mathrm{e}^{-\alpha/n^{3/2}}\right]^{-k}.$$

This can be rewritten as

$$E(\mathrm{e}^{-\alpha \Theta_n}) = \mathrm{e}^{\sqrt{n}\alpha/2}\, 2^{-n}\, (1 + \beta_n)^n,$$

with

$$\beta_n = \varepsilon\, \mathrm{e}^{-\alpha/n^{3/2}} \left[1 - (1-\varepsilon)\, \mathrm{e}^{-\alpha/n^{3/2}}\right]^{-1}.$$

Recall that $\varepsilon = 1/n$. The expansion of β_n reads

$$\beta_n = 1 - \alpha/\sqrt{n} + \alpha^2/n + o(1/n).$$

This implies that

$$E(\mathrm{e}^{-\alpha\Theta_n}) \to \mathrm{e}^{3\alpha^2/8},$$

as $n \to \infty$. This concludes the proof. □

3 Proof of Theorem 1.2

We first describe the law of T_n conditionally on the values taken by L along the path of $(X_k)_k$ before T_n, that is, until the optimal individual $(1, 1, \ldots, 1)$ is hit. This law is the law of a sum of independent geometric random variables, see Lemma 3.3. We deduce the overall law of T_n, see Proposition 3.1. This yields part (i) of the theorem.

As in the previous section, since CLT implies the law of large numbers, we then prove the CLT of part (iii).

We recall that the (1+1) EA, in the LeadingOnes framework, accepts a mutation if and only if the number of leading ones is increased. Hence the dynamics of the Bernoulli flip algorithm proceeds as follows: the chain jumps to a new individual, at time $k+1$, if and only if the leading ones of X_k are left unchanged and its leftmost zero is flipped, no matter which values are taken by the other bits.

Here and later in the paper, $\varepsilon = c/n$ when we deal with algorithms on strings of length n. For all $i \geq 0$, let $p(n,i) = \varepsilon(1-\varepsilon)^i$. As in the one flip framework, $\widetilde{X}_k$ denotes the position of the chain after its k-th jump. We also keep the same definition for σ_k and τ_k.
For all $k \geq 0$, let

$$\ell_k = L(\widetilde{X}_k).$$

Let Y_0 be such that

$$X_0 = (1^{\ell_0}, 0, Y_0).$$

For $k \geq 1$, define Y_k and W_k by

$$\widetilde{X}_k = (1^{\ell_{k-1}}, 1, W_k) = (1^{\ell_k}, 0, Y_k).$$

Lemmas 3.1 and 3.2 below are needed to compute the law of of T_n conditionally on (ℓ_j) in Lemma 3.3.

Lemma 3.1 *(i) For all $k \geq 0$, σ_k depends on the past only through the last score ℓ_{k-1}. That is, the law of σ_k, conditionally on $(X_t)_{\{t<\tau_k\}}$, is the law of σ_k, conditionally on ℓ_{k-1}.*
(ii) For all $k \geq 0$, given $\{\ell_{k-1} = i\}$, the law of σ_k is $\mathcal{G}(p(n,i))$.

Proof. The sojourn time σ_k is the time the algorithm takes to jump from $\widetilde{X}_{k-1}$ to $\widetilde{X}_k$. The leftmost zero of $\widetilde{X}_{k-1}$ is in position $\ell_{k-1}+1$. Thus, one needs to flip the $(\ell_{k-1}+1)$-th bit, while leaving the first ℓ_{k-1} bits unchanged.Thus,

$$P(\sigma_k = t \,|\, \widetilde{X}_0, \ldots, \widetilde{X}_{k-1}) = \varepsilon(1-\varepsilon)^{\ell_{k-1}} \left[1 - \varepsilon(1-\varepsilon)^{\ell_{k-1}}\right]^{(t-1)} .$$

□

Lemma 3.2 *For all $k \geq 1$ let $\widehat{\mathcal{F}}_k = \sigma\{\sigma_i, \ell_j : i \leq k, j \leq k-1\}$, then the law of ℓ_k conditionally on $\widehat{\mathcal{F}}_k$ is the law of ℓ_k conditionally on ℓ_{k-1}.*

Proof. Let P_i denote the probability given $\{\ell_0 = i\}$. Using the strong Markov property:

$$P(\ell_k = i_k | \sigma_k = t_k, \ell_{k-1} = i_{k-1}, \ldots, \sigma_1 = i_1, \ell_0 = i_0) = P(\ell_1 = i_k | \sigma_1 = t_k, \ell_0 = i_{k-1}) \qquad (6)$$

Since $\{\ell_1 = i_1, \sigma_1 = t, \ell_0 = i_0\} = \{L(X_t) = i_1, L(X_{t-1}) = \ldots = L(X_0) = i_0\}$, since $\{\sigma_1 = t, \ell_0 = i_0\} = \{L(X_t) \neq i_0, L(X_{t-1}) = \ldots = L(X_0) = i_0\}$ and using the Markov property on $(L(X_t)_t)$ we derive the following expression:

$$P(\ell_1 = i_k | \sigma_1 = t_k, \ell_0 = i_{k-1}) = \frac{P(L(X_1) = i_k | L(X_0) = i_{k-1})}{P(L(X_1) \neq i_{k-1} | L(X_0) = i_{k-1})}$$

This quantity is indepedent from t_k, hence if we reconsider (6):

$$P(\ell_k = i_k | \sigma_k = t_k, \ell_{k-1} = i_{k-1}, \ldots, \sigma_1 = i_1, \ell_0 = i_0) = P(\ell_k = i_k | \ell_{k-1} = i_{k-1})$$

□

Lemma 3.3 *Conditionally on $\{\ell_0 = i_0, \ldots, \ell_{J-1} = i_{J-1}, \ell_J = n\}$, T_n is the sum of J independent geometric random variables with respective parameters $p(n, i_0), \ldots, p(n, i_{J-1})$.*

Proof. Given the successive LeadingOnes scores $\{\ell_0 = i_0, \ldots, \ell_J = n\}$ until the optimal individual is hit, $T_n = \sum_{k=1}^{J} \sigma_k$. Thus,

$$P(T_n = t | \ell_J = n, \ldots, \ell_0 = i_0) = \sum_{t_1 + \cdots + t_k = t} P(\sigma_J = t_J, \ldots, \sigma_1 = t_1 | \ell_J = n, \ldots, \ell_0 = i_0) \qquad (7)$$

Using Lemmas (3.2) and (3.1) we can derive by induction:

$$P(\ell_J = n, \sigma_J = t_J, \ldots, \sigma_1 = t_1, \ell_0 = i_0) = \prod_{k=1}^{J} P(\ell_k = i_k | \ell_{k-1} = i_{k-1}) P(\sigma_k = t_k | \ell_{k-1} = i_{k-1})$$

Hence, since $\prod_{k=1}^{J} P(\ell_k = i_k | \ell_{k-1} = i_{k-1}) = P(\ell_J = n, \ldots, \ell_0 = i_0)$,

$$P(\sigma_1 = t_1, \ldots, \sigma_J = t_J | \ell_J = n, \ldots, \ell_0 = i_0) = \prod_{k=1}^{J} P(\sigma_k = t_k | \ell_{k-1} = i_{k-1})$$

We can now replace this last equation in (7),

$$P(T_n = t|\ell_J = n, \ldots, \ell_0 = i_0) = \sum_{t_1+\cdots+t_k=t} \prod_{k=1}^{J} P\left(\sigma_k = t_k|\ell_{k-1} = i_{k-1}\right)$$

Let $q(n, i_k)$ denote the probabilty distribution of $\mathcal{G}(p(n, i_k))$.
Then using Lemma 3.1, we can write, as we recognize a product of convolution:

$$P(T_n = t|\ell_J = n, \ldots, \ell_0 = i_0) = q(n, i_{J-1}) * \cdots * q(n, i_0)(t)$$

Thus, given that the search jumps J times until the target $(1, 1, \ldots, 1)$ is hit and given $\{\ell_0 = i_0, \ldots, \ell_J = n\}$, T_n follows the same distribution as the sum of J independent $\mathcal{G}(p(n, i_0))$, $\ldots, \mathcal{G}(p(n, i_{J-1}))$ random variables.

Let us focus on $P(\ell_0 = i_0, \ldots, \ell_{J-1} = i_{J-1}, \ell_J = n)$. In order to compute this quantity, we need the following Lemma:

Lemma 3.4 *Let $k \geq 1$. If X_0 is chosen uniformly in $\{0, 1\}^n$, then, given $\{\ell_{k-1} = i\}$, W_k follows the uniform distribution on $\{0, 1\}^{n-i-1}$:*

$$\mathcal{L}(W_k|\ell_{k-1} = i) = \mathcal{U}(\{0, 1\}^{n-i-1}).$$

Proof. Let us focus on the case where $k = 1$. Let P_i denote the probability conditionally on $\{\ell_0 = i\}$. Let μ denote the probabilty distribution of Y_0 given $\{\ell_0 = i\}$. As X_0 is chosen uniformly in $\{0, 1\}^n$, μ is the uniform distribution on $\{0, 1\}^{n-i-1}$.

$$P_i(W_1 = w) = \sum_{t \geq 1} P_i(\widetilde{X}_1 = (1^i, 1, w), \sigma_1 = t) \tag{8}$$

Since $\{X_{\sigma_1} = (1^i, 1, w), \sigma_1 = t, L(X_0) = i\} = \{X_t = (1^i, 1, w), L(X_{t-1}) = \ldots = L(X_0) = i\}$ and using the Markov property, Equation (8) can be rewritten:

$$P_i(W_1 = w) = \sum_{t \geq 1} P_i(X_1 = (1^i, 1, w)) P_i(L(X_1) = L(X_0))^{t-1}$$

Hence,

$$P_i(W_1 = w) = \frac{P_i(X_1 = (1^i, 1, w))}{P_i(L(X_1) > L(X_0))} \tag{9}$$

We recall that the Markov chain jumps from X_0 to a conformation of higher fitness at time 1 if both none of the ℓ_0 first ones of X_0 are flipped and the leftmost zero of X_0, is flipped. Hence,

$$P_i(L(X_1) > L(X_0)) = \varepsilon(1 - \varepsilon)^i \tag{10}$$

On the other hand, as $P_i(Y_0 = u) = \mu(u) = 1/2^{n-i-1}$,

$$P_i(X_1 = (1^i, 1, w)) = 1/2^{n-i-1} \sum_{u \in \{0,1\}^{n-i-1}} P(X_1 = (1^i, 1, w) | X_0 = (1^i, 0, u)) \tag{11}$$

If $d(w, u)$ denotes the Hamming distance between w and u,

$$P(X_1 = (1^i, 1, w) | X_0 = (1^i, 0, u)) = \varepsilon(1-\varepsilon)^i \varepsilon^{d(w,u)} (1-\varepsilon)^{n-i-1-d(w,u)} \tag{12}$$

Finally Equations (9), (10), (11) and (12) together with

$$\sum_{u \in \{0,1\}^{n-i-1}} \varepsilon^{d(w,u)} (1-\varepsilon)^{n-i-1-d(w,u)} = 1,$$

yield that:

$$P_i(W_1 = w) = 1/2^{n-i-1} \tag{13}$$

Now, using the strong Markov property we can derive the proof for any $k \geq 2$. □

Lemma 3.5 *If X_0 is chosen uniformly at random in the state space, for all $k \geq 1$, the conditional distribution of ℓ_k, given $\{\ell_{k-1} = i_{k-1}\}$, satisfies:*

$$\begin{aligned} P\left(\ell_k = j_k | \ell_{k-1} = j_{k-1}\right) &= 2^{-(j_k - j_{k-1})} \quad \textit{if} \quad i_0 + 1 \leq j_k < n \\ &= 2^{-(n - j_{k-1} - 1)} \quad \textit{if} \quad j_k = n \end{aligned}$$

Proof. This is a direct consequence of Lemma 3.4 □

Now that we know the probability distribution of the sequence of the successive LeadingOnes scores until the target individual $(1, 1, \ldots, 1)$ is hit as well as the distribution of T_n conditional to the values taken by these LeadingOnes scores, we can compute the probability distribution of T_n:

Proposition 3.1 *If X_0 is chosen uniformly at random in $\{0,1\}^n$, then, the probability distribution of T_n satisfies:*

$$P(T_n = t) = \frac{1}{2^n} \sum_J \sum_{\{o \leq i_0 \leq i_1 \leq \ldots \leq i_J = n\}} q(n, i_{J-1}) * \cdots * q(n, i_0)(t) \tag{14}$$

Proof. X_0 being chosen uniformly at random in the search space, $P(\ell_0 = i_0) = \frac{1}{2^{i_0+1}}$. Thus, applying Lemma 3.5,

$$P(\ell_0 = i_0, \ldots, \ell_J = n) = \frac{1}{2^n}$$

The result is then a direct consequence of Lemma 3.3. □

Proof of part (iii). Set $\Theta_n = (T_n - \frac{n^2(e^c-1)}{2c^2})|n^{3/2}$.

Thus,

$$E(\exp(-\alpha\Theta_n)) = \exp\left(\alpha\frac{\sqrt{n}}{2c^2}(e^c-1)\right) E\left(\exp(-\alpha\frac{T_n}{n^{3/2}})\right) \tag{15}$$

According to the distribution of T_n given by (14):

$$E\left(\exp(-\alpha\frac{T_n}{n^{3/2}})\right) = \frac{1}{2^n}\sum_J \sum_{i_0<\cdots<i_{J-1}} E\left(\exp(-\frac{\alpha}{n^{3/2}}(\mathcal{G}(p(n,i_0) + \cdots + \mathcal{G}(p(n,i_{J-1})))\right)$$

Since the variables $(\mathcal{G}(p(n,i_k))_{i_k}$ are independent,

$$\begin{aligned} E\left(\exp(-\alpha\frac{T_n}{n^{3/2}})\right) &= \frac{1}{2^n}\sum_J \sum_{i_0<\cdots<i_{J-1}} \prod_{k=0}^{J-1} E\left(-\frac{\alpha}{n^{3/2}}\mathcal{G}(p(n,i_k)\right) \\ &= \frac{1}{2^n}\prod_{k=0}^{n-1}\left(1+\phi_k(\frac{\alpha}{n^{3/2}})\right) \end{aligned}$$

where $\phi_k(\alpha)$ denotes the Laplace transform of $\mathcal{G}(p(n,i_k))$.

$$\phi_k(\alpha) = \frac{e^{-\alpha}p(n,i_k)}{1-(1-p(n,i_k))e^{-\alpha}}.$$

Recalling that $p(n,i_k) = \varepsilon(1-\varepsilon)^i$ and $\varepsilon = c/n$, we derive that:

$$\prod_{k=0}^{n-1}\left(1+\phi_k(\frac{\alpha}{n^{3/2}})\right) \simeq_{n\to+\infty} 2^n \exp\left(-\alpha\frac{\sqrt{n}}{2c^2}(e^c-1)\right)\exp\left(\frac{3\alpha^2}{8c^3}\frac{(e^{2c}-1)}{2}\right)$$

Thus, replacing this in (15), as n goes to ∞:

$$E(\exp(-\alpha\Theta_n)) \simeq \exp\left(\frac{3\alpha^2}{8c^3}\frac{(e^{2c}-1)}{2}\right)$$

We recognize the Laplace transform of a centered gaussian variable of variance $\frac{3(e^{2c}-1)}{8c^3}$. It ends the proof.

4 Proof of Theorem 1.3

In the MCM$_0$ framework, the algorithm has the possibility to visit several distinct protein conformations (individuals) whose fitness has the same value before it finally jumps to a new individual with higher fitness score. This is not allowed in the $(1+1)_L$-EA where the individual at time k, X_k, is not allowed to jump to an individual X_{k+1} with the same fitness score and that would not be X_k itself.

As in the previous sections we denote by $(\widetilde{X}_k)_k$ the chain defined by the protein conformations taken at the times of fitness jumps $(\tau_k)_k$.

Now, we briefly sketch the proof of Theorem 1.3.

Proof in the one flip case.

Here, we put $\varepsilon = 1/n$.

In the $(1+1)$ framework, once X_0 has been sampled, the path $(\widetilde{X}_k)_k$ becomes entirely deterministic: going trough it means exactly flipping, one at a time, from left to right the zeros of X_0. This is not true in the case of MCM_0 and we cannot adapt directly the proof of Theorem 1.1.

The idea of the proof is close to the one Theorem 1.2. First we consider the law of $\widehat{T}_n$ conditionally on the values taken by L along the path $(X_k)_k$ until the ground state is sampled. The same basics arguments apply. Lemma 3.2 remains true and we find that, as in the one flip $(1+1)$ framework, $(\sigma)_k$ is i.i.d. of law $\mathcal{G}(\varepsilon)$. Now, Lemma 3.3 can easily be adapted:

Lemma 4.1 *Conditionally on $\{\ell_0 = i_0, \ldots, \ell_{J-1} = i_{J-1}, \ell_J = n\}$, $\widehat{T}_n$ is the sum of J i.i.d. geometric random variables with parameter ε, i.e. $\widehat{T}_n$ is negative binomial of parameter (J, ε).*

Lemma 3.4 still holds:

Proof. The proof is a copy the one of Lemma 3.4 up to Equation (9):

$$P_i(W_1 = w) = \frac{P_i(X_1 = (1^i, 1, w))}{P_i(L(X_1) > L(X_0))}$$

In the one flip scenario, a unique bit at a time is flipped during the step of mutation. In consequence, in order to sample $(1^i, 1, w)$ at time 1, we need to have sampled $(1^i, 0, w)$ at time 0. Thus,

$$P_i(X_1 = (1^i, 1, w)) = P(X_1 = (1^i, 1, w) | X_0 = (1^i, 0, w)) P_i(Y_0 = w) = 1/(n 2^{n-i-1}) \tag{16}$$

On an another hand,

$$P_i(L(X_1) > L(X_0)) = 1/n \tag{17}$$

Equalities (16) and (17) applied to Equation (9) return the result for $k = 1$. Finally, the strong Markov property ends the proof for $k > 1$.

Now, we can prove Theorem 1.3 in the one flip case:

Proof of Theorem 1.3 . From the above, we derive the distribution of $\widehat{T}_n$:

$$\begin{aligned} P(\widehat{T}_n = t) &= \frac{1}{2^n} \sum_J \sum_{\{o \le i_0 \le i_1 \le \ldots \le i_J = n\}} \mathcal{NB}(J, \varepsilon)(t) \\ &= \sum_J \frac{1}{2^n} \binom{n}{J} \mathcal{NB}(J, \varepsilon)(t) \end{aligned}$$

We recall from Lemma 2.1 that conditionally on $|X_0| = n - J$, T_n is negative binomial of parameter (J, ε). As X_0 is chosen uniformly at random in $\{0,1\}^n$, it yields that for all $t \ge 0$:

$$P(T_n = t) = P(\widehat{T}_n = t)$$

□

Proof in the Bernoulli flip case.
The dynamics of MCM_0 and $(1+1)_L$ slighlty differ. Though, we notice that in the MCM_0 framework, Lemmas 3.1, 3.2, 3.4, 3.5 still hold. Now, since the proof of Proposition 3.1, is entirely based on these lemmas, we derive that the probabilty distribution of the hitting time is the same in both the MCM_0 and the $(1+1)_L$-EA scenari.

□

5 Conclusion

After examining the two versions (one flip and Bernoulli flip) of the EAs on which we focused, we reach the following conclusion: As $(e^c - 1)/c^2 > 1$ for all $c \in \mathbb{R}_+$, the expected value of the hitting time is higher in the Bernoulli flip than in the one flip. Thus, we can conclude that the one flip performs better than any Bernoulli flip, in terms of the expected hitting time. The same conclusion has already been derived by Garnier et al. [14] for the OneMax problem.

This better performance of the one flip suggests that, despite the ability of the Bernoulli flip to jump from any region of the search space to any other, in a single iteration of the search process, the Bernoulli flip results in a slower convergence to a given individual, in the LeadingOnes framework.

In order to explain this phenomenon, as the Markov chain which models our $(1+1)$ search process accepts a mutation only in case of an increase in the number of leading ones, we should point out the following facts: in the Bernoulli flip framework, the closer the algorithm gets to the target individual, the longer the algorithm waits until it jumps; on the other hand, in the one flip case, the number of leading ones currently present in the bit-string does not interfere in the distribution of the time taken for the search to jump. Also, this probability distribution remains stable as the search draws near to the optimal individual.

References

[1] C. B. Anfinsen. Principles that govern the folding of a protein chain (Nobel lecture). *Science*, 191:223–230, 1973.

[2] T. Bäck. Optimal Mutation Rate in Genetic Search. In S.Forrest, editor, *Proceedings of the 5^{th} International Conference on Genetic Algorithms*, pages 2–8. Morgan Kaufmann, 1993.

[3] A. Bakk, J. S. Høye, A. Hansen, S. Sneppen, and M. H. Jensen. Pathways in two-state protein folding. *Biophys.J.*, 79:2722–2727, 2000.

[4] J. Bérard. Genetic algorithms in random environments: two examples. Preprint.

[5] J. Bérard and A. Bienvenüe. Convergence of a genetic algorithm with finite population. In *Mathematics and computer science (Versailles, 2000)*, pages 155–163. Birkhäuser, Basel, 2000.

[6] J. Bérard and A. Bienvenüe. Un principe d'invariance pour un algorithme génétique en population finie. *C. R. Acad. Sci. Paris Sér. I Math.*, 331(6):469–474, 2000.

[7] K. Binder. *Applications of the Monte Carlo Method in Statistical Physics*. Springer Verlag, Berlin, 1987.

[8] R. Cerf. The dynamics of mutation-selection algorithms with large population sizes. *Ann. Inst. H. Poincaré Probab. Statist.*, 32(4):455–508, 1996.

[9] R. Cerf. Asymptotic convergence of genetic algorithms. *Adv. in Appl. Probab.*, 30(2):521–550, 1998.

[10] K.A. Dill, K. M. Feibig, and H. S. Chan. Cooperativity in protein folding kinetics. *Proc.Natl.Acad.Sci.USA*, 90:1942–1946, 1993.

[11] S. Droste and et al. On the analysis of the (1+1) evolutionary algorithm.

[12] S. Droste, T. Jansen, and I. Wegener. On the analysis of the (1+1) evolutionary algorithm. Technical Report CI-21/98,Univ. Dortmund, Collaborative Research Center 531, 1998.

[13] S. Droste, T. Jansen, and I. Wegener. A rigorous complexity analysis of the $(1+1)$ evolutionary algorithm for linear functions with boolean inputs. In *Proceedings of the Fith IEEE International Conference on Evolutionary Computation*. IEEE Press, 1998.

[14] J. Garnier, L. Kallel, and M. Schoenauer. Rigorous hitting times for binary mutations. *Evolutionary Computation*, 7(2):173–203, 1999.

[15] C. Levinthal. Principles that govern the folding of a protein chain (Nobel lecture). *J. Chem. Phys.*, 65:44–45, 1968.

[16] Christian Mazza and Didier Piau. On the effect of selection in genetic algorithms. *Random Structures Algorithms*, 18(2):185–200, 2001.

[17] P. Del Moral and A. Guionnet. On the stability of interacting processes with applications to filtering and genetic algorithms. *Ann. Inst. H. Poincaré Probab. Statist.*, 37(2):155–194, 2001.

[18] H. Mühlenbein. How genetic algorithms really work: I. mutation and hill-climbing. In R.Manner and B.Manderick, editors, *Proceedings of the* 2^{nd} *Conference on Parallel Problems Solving from Nature*, pages 15–25. Morgan Kaufmann, 1992.

[19] Y. Rabinovich and A. Wigderson. Techniques for bounding the convergence rate of genetic algorithms. *Random Structures Algorithms*, 14(2):111–138, 1999.

[20] G. Rudolph. *Convergence Properties of Evolutionary Algorithms.* Kovac, Hamburg, 1997.

[21] J. A. Schellman. The factors affecting the stability of hydrogen polypeptide structures in solution. *J. Phys. Chem.*, 62, 1485-1494.